

RPG Metrics – Agent

The RPG Metrics Agent performs local scanning of RPG source code and generates local XLSX reports.

The same agent can run in two modes:

- on IBM i/PASE, watching an IFS folder for RPG source stream files
- on Windows, Linux or macOS, watching a local folder of exported RPG source files

RPG source code is scanned locally in the environment where the agent is running.

The agent does not upload RPG source code. If dashboard sync is enabled, only calculated metrics are sent to the RPG Metrics dashboard.

The agent:

- Watches a configured source folder for incoming RPG source files
- Supports IBM i/PASE mode and PC/local-folder mode
- Scans RPG code locally
- Generates an XLSX report locally
- Optionally sends calculated metrics to the RPG Metrics SaaS dashboard
- Moves successfully processed files to **sent/**
- Moves failed or unprocessable files to **unsent/**
- Supports local-only operation with no dashboard token
- Uses dashboard-controlled feature flags when dashboard sync is enabled
- Retrieves active dashboard custom scan rules when enabled
- Can send IBM i completion messages when running on IBM i
- Logs to both a daily log file and the console / **.bat** window

Current Free Feedback Period

RPG Metrics is currently in a free-feedback period while we gather feedback from IBM i developers and teams.

When **DASHBOARD_SYNC=true** and dashboard authentication succeeds, authenticated feedback users currently receive the available Agent feature set, including:

- local scanning
- local XLSX generation
- dashboard sync
- Hotspots worksheet
- RPG Free Compatibility worksheet
- Nesting Analysis worksheet
- Source Analysis nesting markers
- custom scan rules
- team-prefixed scans

When `DASHBOARD_SYNC=false`, the Agent deliberately runs in Free local-only mode. It does not contact the dashboard for authentication, feature flags, custom rules or metrics upload. The local Free report contains the Summary and Source Analysis worksheets only.

If dashboard authentication fails while `DASHBOARD_SYNC=true`, the Agent safely falls back to the same Free local mode.

The Agent is distributed as a versioned ZIP package, for example:

```
rpg-metrics-agent-0.2.4.zip
```

Architecture

IBM i / PASE mode

```
IBM i
|
|  stream file created
|  using PDM, change control, or script
v
RPG Metrics Agent
|
|- scan source locally on IBM i
|- generate XLSX report locally on IBM i
|- optionally request feature flags from RPG Metrics SaaS
|- optionally POST calculated metrics to RPG Metrics SaaS
v
RPG Metrics Dashboard
```

PC / local-folder mode

```
Windows / Linux / macOS
|
|  exported RPG source files placed in a local folder
v
RPG Metrics Agent
|
|- scan source locally on the PC/server
|- generate XLSX report locally
|- optionally request feature flags from RPG Metrics SaaS
|- optionally POST calculated metrics to RPG Metrics SaaS
v
RPG Metrics Dashboard
```

Privacy Model

The agent is designed for teams that do not want to upload source code.

The agent does not upload RPG source code.

If dashboard sync is enabled, only calculated metrics are transmitted to the RPG Metrics dashboard, such as:

- total lines
- executable lines
- comment percentage
- cyclomatic complexity
- maintainability index
- RPG Free compatibility counts
- procedure and subroutine counts
- cyclomatic hotspot summaries

The XLSX report is generated locally and remains in the environment where the agent is running.

In IBM i mode, RPG source and XLSX reports remain on IBM i.

In PC/local-folder mode, the agent scans RPG source files that have already been exported to the local PC or server. Those source files are still not uploaded to the dashboard.

Folder Layout

IBM i example

A typical IBM i installation uses two main locations:

```
/home/metrics-agent/  
  agent.py  
  agent.conf  
  agent.conf.example  
  requirements.txt  
  
  config/  
  metrics/  
  reporting/  
  reports/  
  logs/  
  .venv/  
  
/home/devuser/metrics-sources/  
  sent/  
  unsent/
```

The agent code lives in:

```
/home/metrics-agent/
```

The watched source stream file folder is normally configured using `IBMI_IFS_PATH` or `SCAN_PATH` in `agent.conf`.

Example:

```
SCAN_PATH=  
IBMI_IFS_PATH=/home/devuser/metrics-sources
```

Generated reports are stored under `REPORTS_PATH`.

Example:

```
REPORTS_PATH=/home/metrics-agent/reports
```

Windows / PC example

A typical Windows test installation can use:

```
C:\rpg-metrics-agent\  
  agent.py  
  agent.conf  
  agent.conf.example  
  run-agent.bat  
  requirements.txt  
  
  config\  
  metrics\  
  reporting\  
  sources\  
    sent\  
    unsent\  
  reports\  
  logs\  
  .venv\  

```

Example Windows paths:

```
SCAN_PATH=C:/rpg-metrics-agent/sources  
REPORTS_PATH=C:/rpg-metrics-agent/reports
```

Forward slashes are valid in Python on Windows and avoid escaping issues.

Folder Purpose

Folder	Purpose
/home/metrics-agent/	IBM i agent code and configuration
/home/metrics-agent/reporting/	XLSX report writer code
/home/metrics-agent/reports/	Generated local XLSX reports on IBM i
/home/metrics-agent/logs/	IBM i agent log files
/home/devuser/metrics-sources/	Incoming IBM i source stream files
/home/devuser/metrics-sources/sent/	Successfully processed source files
/home/devuser/metrics-sources/unsent/	Files that failed local scan, report generation, or upload validation
C:\rpg-metrics-agent\sources\	Example Windows source folder
C:\rpg-metrics-agent\reports\	Example Windows reports folder
C:\rpg-metrics-agent\logs\	Example Windows logs folder

Python Requirements

The RPG Metrics Agent is designed to run using Python 3.9 or later.

Recommended version:

```
Python 3.9
```

Required Python packages are listed in:

```
requirements.txt
```

Current requirements:

```
requests  
openpyxl
```

The agent intentionally avoids heavy compiled dependencies such as:

```
numpy  
pandas  
matplotlib
```

This keeps installation simpler on IBM i.

IBM i Installation

1. Copy the agent to IBM i

Copy the `rpg-metrics-agent` folder to the IFS:

```
/home/metrics-agent
```

A typical ZIP extract should result in:

```
/home/metrics-agent/  
  agent.py  
  agent.conf.example  
  requirements.txt  
  config/  
  metrics/  
  reporting/  
  README.md  
  VERSION.txt
```

2. Create the source input folder

Create the folder that will receive source stream files:

```
mkdir -p /home/devuser/metrics-sources  
mkdir -p /home/devuser/metrics-sources/sent  
mkdir -p /home/devuser/metrics-sources/unsent
```

Create the local report and log folders:

```
mkdir -p /home/metrics-agent/reports  
mkdir -p /home/metrics-agent/logs
```

3. Create Python virtual environment

Use Python 3.9 from the IBM i open source packages.

Python 3.9 can be installed using IBM i ACS Open Source Package Management.

The 0.2.4 Agent must be run with Python 3.9.x.

Install the following IBM i open source packages:

```
python39  
python39-pip
```

Do not remove older Python versions unless you know they are unused by other applications.

Create the virtual environment:

```
cd /home/metrics-agent  
  
rm -rf .venv  
  
/QOpenSys/pkgsrc/bin/python3.9 -m venv .venv  
source .venv/bin/activate
```

Confirm the Python version:

```
python --version
```

Expected output:

```
Python 3.9.x required
```

Python Version

RPG Metrics Agent requires **Python 3.9**.

The current Agent package has been tested with:

- Python 3.9.21 on IBM i/PASE
- Python 3.9 on Windows/Linux/macOS

Some protected Agent modules are distributed as compiled Python bytecode (**.pyc**), so the Python runtime must be compatible with the version used to build the package.

We recommend using Python 3.9.x for the 0.2.4 Agent release.

4. Install dependencies

On IBM i, disable pip's progress bar to avoid terminal/threading issues in PASE:

```
python -m pip install --progress-bar off -r requirements.txt
```

If needed, install without cache:

```
python -m pip install --progress-bar off --no-cache-dir requests openpyxl
```

Verify the dependencies:

```
python -c "import requests; import openpyxl; print('deps ok')"  
python -c "from metrics.engine import scan_stream_file; print('scanner ok')"  
python -c "from reporting.xlsx_report import generate_excel_report; print('xlsx  
writer ok')"
```

Windows / PC Local-Folder Mode

The same agent can also run on Windows against a folder of exported RPG source files.

This is useful for testing, demonstrations, or sites where RPG source has already been exported from IBM i.

1. Create the folder layout

Example:

```
C:\rpg-metrics-agent\  
  agent.py  
  agent.conf  
  agent.conf.example  
  run-agent.bat  
  requirements.txt  
  
  config\  
  metrics\  
  reporting\  
  sources\  
    sent\  
    unsent\  
  reports\  
  logs\  
  .venv\  

```

Create the main folders:

```
mkdir C:\rpg-metrics-agent
mkdir C:\rpg-metrics-agent\sources
mkdir C:\rpg-metrics-agent\sources\sent
mkdir C:\rpg-metrics-agent\sources\unsent
mkdir C:\rpg-metrics-agent\reports
mkdir C:\rpg-metrics-agent\logs
```

2. Create the virtual environment

```
cd C:\rpg-metrics-agent

py -3.9 -m venv .venv
.\.venv\Scripts\python -m pip install -r requirements.txt
```

If `py -3.9` is not available, use:

```
python -m venv .venv
.\.venv\Scripts\python -m pip install -r requirements.txt
```

3. Example PC configuration

```
[agent]
PLATFORM_MODE=pc
DASHBOARD_SYNC=false

SCAN_PATH=C:/rpg-metrics-agent/sources
REPORTS_PATH=C:/rpg-metrics-agent/reports

IBMI_IFS_PATH=/home/devuser/metrics-sources

TEAM_NAME=Personal
POLL_INTERVAL=60

IBMI_MESSAGE_ENABLED=false
IBMI_MESSAGE_FALLBACK_USER=

TOKEN_URL=https://kdp-software.co.uk/rpg-metrics/api/token/refresh/
PUSH_URL=https://kdp-software.co.uk/rpg-metrics/api/ibmi/metrics/
FEATURES_URL=https://kdp-software.co.uk/rpg-metrics/api/ibmi/agent/features/
RULES_URL=https://kdp-software.co.uk/rpg-metrics/api/ibmi/agent/rules/
AUTH_REFRESH_TOKEN=

RETRY_MAX=5
```

```
RETRY_BACKOFF_BASE=1  
RETRY_BACKOFF_CAP=60
```

4. Run manually on Windows

```
cd C:\rpg-metrics-agent  
.\.venv\Scripts\python agent.py
```

Or run using:

```
.\run-agent.bat
```

Configuration

The repository includes a sample configuration file:

```
agent.conf.example
```

Create your local configuration file.

On IBM i:

```
cd /home/metrics-agent  
cp agent.conf.example agent.conf
```

On Windows:

```
cd C:\rpg-metrics-agent  
Copy-Item .\agent.conf.example .\agent.conf
```

Edit:

```
agent.conf
```

IBM i example

[agent]

```
PLATFORM_MODE=auto
DASHBOARD_SYNC=false

SCAN_PATH=
IBMI_IFS_PATH=/home/devuser/metrics-sources
REPORTS_PATH=/home/metrics-agent/reports

TOKEN_URL=https://kdp-software.co.uk/rpg-metrics/api/token/refresh/
PUSH_URL=https://kdp-software.co.uk/rpg-metrics/api/ibmi/metrics/
FEATURES_URL=https://kdp-software.co.uk/rpg-metrics/api/ibmi/agent/features/
RULES_URL=https://kdp-software.co.uk/rpg-metrics/api/ibmi/agent/rules/
AUTH_REFRESH_TOKEN=

TEAM_NAME=Personal
POLL_INTERVAL=60

IBMI_MESSAGE_ENABLED=false
IBMI_MESSAGE_FALLBACK_USER=QSYSOPR

RETRY_MAX=5
RETRY_BACKOFF_BASE=1
RETRY_BACKOFF_CAP=60
```

PC example

[agent]

```
PLATFORM_MODE=pc
DASHBOARD_SYNC=false

SCAN_PATH=C:/rpg-metrics-agent/sources
REPORTS_PATH=C:/rpg-metrics-agent/reports

IBMI_IFS_PATH=/home/devuser/metrics-sources

TEAM_NAME=Personal
POLL_INTERVAL=60

IBMI_MESSAGE_ENABLED=false
IBMI_MESSAGE_FALLBACK_USER=

TOKEN_URL=https://kdp-software.co.uk/rpg-metrics/api/token/refresh/
PUSH_URL=https://kdp-software.co.uk/rpg-metrics/api/ibmi/metrics/
FEATURES_URL=https://kdp-software.co.uk/rpg-metrics/api/ibmi/agent/features/
RULES_URL=https://kdp-software.co.uk/rpg-metrics/api/ibmi/agent/rules/
AUTH_REFRESH_TOKEN=

RETRY_MAX=5
```

```
RETRY_BACKOFF_BASE=1
RETRY_BACKOFF_CAP=60
```

Configuration Values

Setting	Purpose
PLATFORM_MODE	Controls runtime mode. Use auto , ibmi , or pc . auto detects IBM i/PASE where possible, otherwise uses PC/local-folder mode
SCAN_PATH	Neutral source folder watched by the agent. Takes precedence over IBMI_IFS_PATH when set
IBMI_IFS_PATH	IBM i/PASE source folder fallback. Kept for compatibility with existing IBM i installs
DASHBOARD_SYNC	Set to false for local-only XLSX reports, or true to send calculated metrics to the RPG Metrics dashboard
REPORTS_PATH	Folder where local XLSX reports are written
TOKEN_URL	Endpoint used to exchange the refresh token for an access token
PUSH_URL	Endpoint used to upload calculated metrics
FEATURES_URL	Endpoint used to retrieve dashboard-controlled feature flags
RULES_URL	Endpoint used to retrieve active dashboard custom scan rules
AUTH_REFRESH_TOKEN	Long-lived agent refresh token from the RPG Metrics profile page. Required only when DASHBOARD_SYNC=true ; leave blank for local-only use
TEAM_NAME	Default team name when no team prefix is used
POLL_INTERVAL	Seconds between folder scans
IBMI_MESSAGE_ENABLED	Whether IBM i completion messages are sent. Only applies when PLATFORM_MODE resolves to ibmi
IBMI_MESSAGE_FALLBACK_USER	IBM i profile notified only when the stream-file owner cannot be resolved
RETRY_MAX	Maximum retry attempts for transient network/server issues
RETRY_BACKOFF_BASE	Base delay for retry backoff
RETRY_BACKOFF_CAP	Maximum retry delay

Authentication - Agent Token

A refresh token is only required when:

```
DASHBOARD_SYNC=true
```

No token is required for local-only XLSX report generation.

Get your token

1. Log in to RPG Metrics:

```
https://kdpsoftware.co.uk/rpg-metrics/
```

2. Go to your Profile page

3. Click:

Generate Agent Token

4. Copy the generated token

Configure the agent

Update your `agent.conf`:

```
AUTH_REFRESH_TOKEN=<your token>
```

Notes

- The Profile page generates a refresh token
- The agent uses the refresh token to request short-lived access tokens
- No user password is stored in the agent
- Tokens can be regenerated at any time from the Profile page
- If a refresh token expires or is replaced, update `agent.conf` and restart the agent

Dashboard Feature Flags

When `DASHBOARD_SYNC=true`, the agent contacts the RPG Metrics dashboard to retrieve feature flags for the authenticated user.

The dashboard controls:

- whether the agent is enabled
- whether local XLSX reports are enabled
- whether dashboard sync is enabled

- whether Hotspots worksheets are included
- whether RPG Free Compatibility worksheets are included
- whether Nesting Analysis is enabled
- whether custom scan rules are enabled
- whether team features are enabled

During the current free-feedback period, all available Agent features are enabled for authenticated dashboard-connected users.

When standard subscription controls apply, the dashboard feature flags determine which plan-controlled worksheets and features are available.

When `DASHBOARD_SYNC=false`, the agent does not contact the dashboard for feature flags, custom rules, or metrics upload.

Typical local-only feature log:

```
Dashboard sync disabled in agent.conf. Running in local-only mode with no
dashboard authentication, no feature-flag request, no custom-rule request and no
metrics upload.
Agent startup feature mode: platform_mode=pc, subscription=free-local,
dashboard_sync=False, local_xlsx=True, hotspots=False, rpgfree=False,
nesting=False, custom_rules=False
```

Custom Scan Rules

The agent includes built-in RPG decision markers, executable keywords and RPG Free compatibility checks.

When connected to an RPG Metrics dashboard account, the agent can also retrieve the user's active custom rules from User Settings.

Custom rules are added to the built-in defaults; they do not replace them.

If the dashboard is unavailable, or custom rules are not enabled for the account, the agent continues using its embedded defaults.

The agent fetches rules at startup, so users need to restart it after changing rules in the dashboard.

IBM i Completion Notifications

The agent can optionally send an IBM i message when a submitted source file reaches a final outcome.

Notifications are sent to the owner of the submitted stream file. This allows the agent to run under a service profile while notifying the IBM i user who created the stream file with `CPYTOSTMF`.

IBM i completion notifications only apply when `PLATFORM_MODE` resolves to `ibmi`.

Enable notifications in `agent.conf`:

```
IBMI_MESSAGE_ENABLED=true  
IBMI_MESSAGE_FALLBACK_USER=QSYSOPR
```

The fallback profile is used only when the owner of a submitted stream file cannot be resolved.

Successful notifications include the source file name, cyclomatic complexity and maintainability index.

Files moved to **unsent/** generate a failure notification.

Retryable dashboard or network errors do not generate a final failure message because the source remains queued for retry.

Example message:

```
RPG Metrics complete: LMARK_LAN8XAPPS_QRPGLESRC_MITMCLAS.RPGLE.  
XLSX created and dashboard synced.  
CC=43, MI=72.73.
```

Safe Fallback Mode

If dashboard authentication or feature retrieval fails, the Agent falls back to a safe Free local mode:

- local scan enabled
- basic XLSX report enabled
- dashboard sync disabled
- Hotspots worksheet disabled
- RPG Free Compatibility worksheet disabled
- Nesting Analysis disabled
- nesting calculation not performed
- team/trend/custom-rule features disabled

This ensures the user still receives a local report without the Agent assuming access to paid or dashboard-controlled features.

When **DASHBOARD_SYNC=false**, the Agent uses Free local-only mode immediately and does not attempt dashboard authentication.

Subscription Behaviour

Feature behaviour is controlled by the RPG Metrics dashboard.

Current free-feedback period

During the current free-feedback period, all available features are enabled for authenticated dashboard-connected users.

Standard subscription behaviour

When the current free-feedback period ends, subscription behaviour is:

Feature	Free	Basic	Pro
IBM i / PC agent	Yes	Yes	Yes
Local scan	Yes	Yes	Yes
Local basic XLSX	Yes	Yes	Yes
Dashboard sync	First 30 days from registration	Yes	Yes
Hotspots worksheet	No	Yes	Yes
RPG Free Compatibility worksheet	No	Yes	Yes
Nesting Analysis	No	Yes	Yes
Custom scan rules	No	Yes	Yes
Team features	No	Limited or no	Yes
Trend/team reporting	No	Limited	Yes

For Free users after the 30-day dashboard sync period, the agent is expected to continue generating the basic local XLSX report.

File Naming Convention

The agent supports personal and team-prefixed source files.

Personal scan

```
LIBRARY_SOURCEFILE_MEMBER.TYPE
```

Example:

```
KEV_QRPGLESRC_BFSALES.RPGLE
```

This means:

```
LIBRARY      = KEV
SOURCEFILE   = QRPGLSRC
MEMBER       = BFSALES
TYPE         = RPGLE
TEAM         = value from TEAM_NAME in agent.conf
```

Team scan

```
TEAM_LIBRARY_SOURCEFILE_MEMBER.TYPE
```

Example:

```
SETT5_KEV_QRPGLSRC_BFSALES.RPGLE
```

This means:

```
TEAM        = SETT5
LIBRARY      = KEV
SOURCEFILE   = QRPGLSRC
MEMBER       = BFSALES
TYPE         = RPGLE
```

If a team prefix is present, it overrides the configured `TEAM_NAME`.

Team names must not contain underscores.

Supported Source Types

The file extension is used as the program type.

Supported values include:

```
RPG
SQLRPG
RPGLE
SQLRPGLE
RPGFREE
```

Examples:

```
KEV_QRPGLESRC_BFSALES.RPGLE  
SETT5_SETTURN5S_QRPGSRC_TAEXT1.RPG
```

Creating Stream Files on IBM i

The agent expects source to arrive as stream files in the configured IFS folder.

Example PDM user option:

```
CPYTOSTMF  
  FROMMBR('/qsys.lib/&l.lib/&f.file/&n.mbr')  
  TOSTMF('/home/devuser/metrics-sources/&l_&f_&n.&t')  
  STMFOPT(*REPLACE)  
  CVTDTA(*AUTO)  
  DBFCCSID(*FILE)  
  STMFCCSID(*STDASCII)
```

This converts a source member to a stream file for scanning.

For team scans, prefix the stream file name with the team name:

```
CPYTOSTMF  
  FROMMBR('/qsys.lib/&l.lib/&f.file/&n.mbr')  
  TOSTMF('/home/devuser/metrics-sources/SETT5_&l_&f_&n.&t')  
  STMFOPT(*REPLACE)  
  CVTDTA(*AUTO)  
  DBFCCSID(*FILE)  
  STMFCCSID(*STDASCII)
```

PC Source Files

In PC/local-folder mode, the agent expects RPG source files to already exist as text files in the configured `SCAN_PATH`.

The source file name should still follow the same naming convention:

```
LIBRARY_SOURCEFILE_MEMBER.TYPE  
TEAM_LIBRARY_SOURCEFILE_MEMBER.TYPE
```

Example:

```
KEV_QRPGLESRC_BFSALES.RPGLE  
SETT5_KEV_QRPGLESRC_BFSALES.RPGLE
```

The agent does not extract source directly from IBM i in PC mode. It scans files that have already been exported or copied to the local folder.

Running the Agent

To verify the agent is working, place a test source file in the configured source folder.

Use `SCAN_PATH` when it is set. If `SCAN_PATH` is blank, the agent falls back to `IBMI_IFS_PATH`.

The agent will scan the file and move it to the `sent` folder if successful.

Run manually on IBM i

```
cd /home/metrics-agent  
source .venv/bin/activate  
python agent.py
```

Run manually on Windows

```
cd C:\rpg-metrics-agent  
.\.venv\Scripts\python agent.py
```

Or use:

```
.\run-agent.bat
```

Typical Log Output

Windows local-only mode

```
BASE_DIR = C:\rpg-metrics-agent  
CONF_PATH = C:\rpg-metrics-agent\agent.conf  
LOG_FILE = C:\rpg-metrics-agent\logs\agent_20260716.log  
PLATFORM_MODE_CONFIGURED = pc  
PLATFORM_MODE = pc  
SCAN_PATH = C:\rpg-metrics-agent\sources  
IFS_PATH = C:\rpg-metrics-agent\sources
```

```

REPORTS_PATH = C:\rpg-metrics-agent\reports
DASHBOARD_SYNC = False
Starting RPG Metrics Agent with local XLSX generation and feature flags
Dashboard sync disabled in agent.conf. Running in local-only mode with no
dashboard authentication, no feature-flag request, no custom-rule request and no
metrics upload.
Agent startup feature mode: platform_mode=pc, subscription=free-local,
dashboard_sync=False, local_xlsx=True, hotspots=False, rpgfree=False,
nesting=False, custom_rules=False
Scan cycle started. Watching: C:\rpg-metrics-agent\sources
No source files waiting in scan path. Next check in 60 seconds.
Sleeping for 60 seconds before next scan cycle.

```

IBM i dashboard-sync mode

```

BASE_DIR = /home/metrics-agent
CONF_PATH = /home/metrics-agent/agent.conf
LOG_FILE = /home/metrics-agent/logs/agent_20260716.log
PLATFORM_MODE_CONFIGURED = auto
PLATFORM_MODE = ibmi
SCAN_PATH = (not set; using IBMI_IFS_PATH)
IFS_PATH = /home/devuser/metrics-sources
REPORTS_PATH = /home/metrics-agent/reports
DASHBOARD_SYNC = True
Starting RPG Metrics Agent with local XLSX generation and feature flags
Dashboard sync enabled in agent.conf. Dashboard authentication required.
Access token obtained successfully
Agent features loaded: subscription=beta, local_xlsx=True, dashboard_sync=True,
hotspots=True, rpgfree=True, nesting=True, team=True
Agent startup feature mode: platform_mode=ibmi, subscription=beta,
dashboard_sync=True, local_xlsx=True, hotspots=True, rpgfree=True, nesting=True,
custom_rules=True
Scan cycle started. Watching: /home/devuser/metrics-sources

```

Running as an IBM i Subsystem Job

Example CL wrapper:

```

PGM

/* Path to the Python virtual environment and agent */
DCL      VAR(&CMD) TYPE(*CHAR) LEN(512) VALUE('source +
        /home/metrics-agent/.venv/bin/activate && +
        python /home/metrics-agent/agent.py')

/* Execute under Qshell */
QSH      CMD(&CMD)

```

```
ENDPGM
```

Attach this program to a subsystem job to keep the agent running continuously.

Example IBM i Subsystem Setup

The agent can be run as a controlled IBM i subsystem job.

The example below creates a test subsystem so you can validate the setup without replacing an existing live subsystem.

This example uses:

```
Subsystem: KDPQGPL/RPGMAGTST
Job queue: KDPQGPL/RPGMAGTST
Class:     KDPQGPL/RPGMAGTST
Program:   KDPQGPL/RPGMAGENT
User:      KEV
Job name:  RPGMAGTST
```

The launcher program name remains **KDPQGPL/RPGMAGENT**.

The subsystem, job queue, class, and job name are changed to **RPGMAGTST**.

Create test subsystem CL program

Create a source member such as:

```
KDPQGPL/QCLSRC member CRTMAGTST
```

Use this CL source:

```
PGM

/* ===== */
/* RPG Metrics Agent TEST subsystem setup           */
/* Creates test subsystem KDPQGPL/RPGMAGTST          */
/* Keeps launcher program as KDPQGPL/RPGMAGENT       */
/* ===== */

CRTJOBQ    JOBQ(KDPQGPL/RPGMAGTST) +
           TEXT('RPG Metrics Agent TEST job queue')

CRTCLS     CLS(KDPQGPL/RPGMAGTST) +
           RUNPTY(50) +
```

```

        TIMESLICE(2000) +
        PURGE(*YES) +
        DFTWAIT(30) +
        CPUTIME(*NOMAX) +
        MAXTMPSTG(*NOMAX) +
        MAXTHD(*NOMAX) +
        TEXT('RPG Metrics Agent TEST class')

CRTSBSD  SBSD(KDPQGPL/RPGMAGTST) +
        POOLS((1 *BASE)) +
        MAXJOBS(*NOMAX) +
        SGNDSPF(QSYS/QDSIGNON) +
        TEXT('RPG Metrics Agent TEST subsystem')

ADDJOBQE SBSD(KDPQGPL/RPGMAGTST) +
        JOBQ(KDPQGPL/RPGMAGTST) +
        SEQNBR(10) +
        MAXACT(1)

ADDPJE   SBSD(KDPQGPL/RPGMAGTST) +
        PGM(KDPQGPL/RPGMAGENT) +
        USER(KEV) +
        STRJOBS(*YES) +
        INLJOBS(1) +
        THRESHOLD(1) +
        ADLJOBS(0) +
        MAXJOBS(1) +
        JOB(RPGMAGTST) +
        JOBD(*USRPRF) +
        MAXUSE(*NOMAX) +
        WAIT(*YES) +
        POOLID(1) +
        CLS(KDPQGPL/RPGMAGTST)

ENDPGM

```

Compile it:

```

CRTBNDCL  PGM(KDPQGPL/CRTMAGTST) +
          SRCFILE(KDPQGPL/QCLSRC) +
          SRCMBR(CRTMAGTST) +
          TEXT('Create RPG Metrics test subsystem')

```

Run it:

```
CALL KDPQGPL/CRTMAGTST
```

Start the test subsystem:

```
STRSBS SBSD(KDPQGPL/RPGMAGTST)
```

Check the running job:

```
WRKACTJOB SBS(RPGMAGTST)
```

Stop the test subsystem:

```
ENDSBS SBS(RPGMAGTST) OPTION(*CNTRLD) DELAY(30)
```

Cleanup test subsystem

Create a separate cleanup CL member if required:

```
PGM

ENDSBS  SBS(RPGMAGTST) OPTION(*IMMED)
DLTSBSD SBSD(KDPQGPL/RPGMAGTST)
DLTJOBQ JOBQ(KDPQGPL/RPGMAGTST)
DLTCLS  CLS(KDPQGPL/RPGMAGTST)

ENDPGM
```

Do not run the live subsystem and the test subsystem at the same time if both use the same `/home/metrics-agent/agent.py` and the same `agent.conf`.

Both jobs would watch the same configured source folder.

Polling Interval

The agent checks the configured source folder every `POLL_INTERVAL` seconds.

Example:

```
POLL_INTERVAL=60
```

A value of `60` means a newly created stream file may wait up to 60 seconds before being picked up.

For testing, a lower value can be used.

For production, `60` seconds is usually a reasonable balance.

XLSX Reports

For every scan, the agent can create a local XLSX report in the configured `REPORTS_PATH`.

Reports are organised by:

```
reports/<team>/<library>/<source-file>/
```

IBM i example:

```
/home/metrics-agent/reports/SETT5/SETTURN5S/QRPGSRC/
```

Windows example:

```
C:\rpg-metrics-agent\reports\SETT5\SETTURN5S\QRPGSRC\
```

Example report file:

```
SETT5_SETTURN5S_QRPGSRC_TAEXT1_RPG_20260514_103115.xlsx
```

Summary sheet

The Summary sheet includes:

- team
- library
- source file
- member
- code type
- scanned timestamp
- subscription level
- dashboard sync status
- total lines
- comment lines
- executable lines
- comment percentage
- cyclomatic complexity
- maintainability index
- RPG Free compatibility counts
- procedure count
- subroutine count

- hotspot count
- maximum nesting depth, routine and line when Nesting Analysis is enabled

Source Analysis sheet

The Source Analysis sheet includes line-level analysis:

- line number
- source line
- code type
- blank line flag
- comment flag
- executable flag
- decision point flag
- RPG Free compatibility flag

When Nesting Analysis is enabled, the Source Analysis sheet also includes a **Nesting** column with **X** and **E** markers showing nesting block starts and ends.

Nesting Analysis sheet

The Nesting Analysis sheet identifies routine-level nesting depth and highlights deeply nested RPG logic.

When enabled, nesting is calculated locally during the source scan. The report includes:

- maximum nesting depth
- line where the maximum depth occurs
- routine containing the maximum depth
- routine-level nesting hotspot information
- **X** and **E** markers in the Source Analysis sheet showing nesting block starts and ends

Nesting Analysis is available with Basic and Pro subscriptions.

It is not calculated in Free local-only mode or authentication-fallback mode.

Hotspots sheet

The Hotspots sheet lists cyclomatic complexity hotspots.

During the current free-feedback period, this sheet is enabled.

When standard subscription controls apply, this sheet is controlled by subscription feature flags.

RPG Free Compatibility sheet

The RPG Free Compatibility sheet lists lines identified as potential RPG Free conversion blockers.

During the current free-feedback period, this sheet is enabled.

When standard subscription controls apply, this sheet is controlled by subscription feature flags.

Error Handling

The agent handles three broad categories of errors.

Local scan failure

If the local scan fails, the source file is moved to:

```
unsent/
```

A sidecar reason file is written.

Example:

```
MYLIB_QRPGLESRC_BADFILE.RPGLE.reason.txt
```

Local XLSX generation failure

If report generation fails, the source file is moved to:

```
unsent/
```

A reason file is written.

Dashboard/server/network issue

If the dashboard upload fails because of a transient server or network issue, the file is left in place for a future retry.

If the dashboard rejects the payload with a permanent client-side validation error, the file is moved to:

```
unsent/
```

A reason file is written.

Logs

Logs are written to the **logs/** folder under the agent installation folder.

The latest agent also writes log output to the console, so a Windows **.bat** window does not appear frozen.

IBM i example:

```
/home/metrics-agent/logs/agent_20260716.log
```

Windows example:

```
C:\rpg-metrics-agent\logs\agent_20260716.log
```

Useful IBM i check:

```
tail -100 /home/metrics-agent/logs/agent_20260716.log
```

Useful Windows check:

```
Get-Content C:\rpg-metrics-agent\logs\agent_20260716.log -Tail 100 -Wait
```

Check which agent process is running on IBM i:

```
ps -ef | grep agent.py | grep -v grep
```

Troubleshooting

Check Python

IBM i:

```
cd /home/metrics-agent  
source .venv/bin/activate  
python --version
```

Windows:

```
cd C:\rpg-metrics-agent  
.\.venv\Scripts\python --version
```

Expected:

```
Python 3.9.x
```

Check dependencies

```
python -c "import requests; import openpyxl; print('deps ok')"
```

On Windows:

```
.\.venv\Scripts\python -c "import requests; import openpyxl; print('deps ok')"
```

Check scanner import

```
python -c "from metrics.engine import scan_stream_file; print('scanner ok')"
```

On Windows:

```
.\.venv\Scripts\python -c "from metrics.engine import scan_stream_file; print('scanner ok')"
```

Check XLSX writer import

```
python -c "from reporting.xlsx_report import generate_excel_report; print('xlsx writer ok')"
```

On Windows:

```
.\.venv\Scripts\python -c "from reporting.xlsx_report import generate_excel_report; print('xlsx writer ok')"
```

Check feature endpoint

From another machine, using an access token:

```
Invoke-RestMethod `
    -Method Get `
```

```
-Uri "https://kdp-software.co.uk/rpg-metrics/api/ibmi/agent/features/" `
-Headers @{ Authorization = "Bearer <ACCESS_TOKEN>" }
```

A successful response includes feature-control fields such as:

```
agent_enabled           : True
local_xlsx_enabled      : True
dashboard_sync_enabled  : True
xlsx_hotspots_enabled   : True
xlsx_rpgfree_analysis_enabled : True
xlsx_nesting_enabled    : True
team_enabled            : True
```

The exact enabled values depend on the current account feature entitlements.

Check deployed agent version

On IBM i:

```
grep -n "PLATFORM_MODE" /home/metrics-agent/agent.py
grep -n "SCAN_PATH" /home/metrics-agent/agent.py
grep -n "FEATURES_URL" /home/metrics-agent/agent.py
grep -n "fetch_agent_features" /home/metrics-agent/agent.py
grep -n "Agent startup feature mode" /home/metrics-agent/agent.py
```

On Windows:

```
Select-String -Path C:\rpg-metrics-agent\agent.py -Pattern
"PLATFORM_MODE|SCAN_PATH|FEATURES_URL|fetch_agent_features|Agent startup feature
mode"
```

Security Model

RPG Metrics is designed to ensure that sensitive IBM i source information is protected.

Source Code Protection

- RPG source code is scanned locally in the environment where the agent runs
- In IBM i mode, source code and XLSX reports remain on IBM i
- In PC/local-folder mode, source code and XLSX reports remain on the local PC/server
- RPG source code is not transmitted to the RPG Metrics dashboard by the metrics-only agent flow

Data Transmission

Only calculated metrics are transmitted to the RPG Metrics dashboard.

All communication between the agent and the RPG Metrics platform uses HTTPS.

Metrics include values such as:

- cyclomatic complexity
- maintainability index
- comment percentage
- line counts
- hotspot summaries
- RPG Free compatibility counts

Encryption of Sensitive Data

Sensitive identifiers are encrypted at rest in the RPG Metrics database, including:

- library names
- source file names
- program names

Encryption is scoped per:

- user, for personal scans
- team, for shared team scans

Sensitive identifiers are decrypted only when required for authorised users to view their own personal or team data.

Email Handling

User email addresses are encrypted at rest where supported by the platform.

Email addresses may be decrypted only for platform communication, such as account, token, report, or notification emails.

What This Means

- Your source code remains in the environment where the agent runs
- In IBM i mode, source code and XLSX reports remain on IBM i
- In PC/local-folder mode, source code and XLSX reports remain on the local PC/server
- Only calculated metrics are sent to the dashboard
- Sensitive object identifiers are protected at rest
- The platform provides code quality insights without storing readable source code

Integration with Change Control Tools

RPG Metrics can be integrated into existing IBM i development workflows by triggering scans automatically during change promotion.

This allows teams to identify complexity issues and RPG Free conversion blockers before code reaches testing or production environments.

The agent works with any change control system capable of executing a command after a promotion step.

Examples include:

- MDCMS
- TurnOver
- Aldon
- Implementer
- custom deployment scripts

Typical IBM i Workflow

```
Developer updates RPG source
  |
  v
Change control promotion
  |
  v
Post-promotion command executes CPYTOSTMF
  |
  v
Stream file written to metrics-agent source folder
  |
  v
RPG Metrics Agent scans code automatically
  |
  |- XLSX report generated locally
  |- Metrics optionally sent to RPG Metrics dashboard
  v
Dashboard history updated
```

Example Post-Promotion Command

A promotion step can convert the source member into a stream file using **CPYTOSTMF**.

Personal example:

```
CPYTOSTMF
FROMMBR('/qsys.lib/&l.lib/&f.file/&n.mbr')
TOSTMF('/home/devuser/metrics-sources/&l_&f_&n.&t')
STMFOPT(*REPLACE)
CVTDTA(*AUTO)
```

```
DBFCCSID(*FILE)
STMFCSSID(*STDASCII)
```

Team example:

```
CPYTOSTMF
FROMMBR(' /qsys.lib/&l.lib/&f.file/&n.mbr')
TOSTMF('/home/devuser/metrics-sources/SETT5_&l_&f_&n.&t')
STMFOPT(*REPLACE)
CVTDTA(*AUTO)
DBFCCSID(*FILE)
STMFCSSID(*STDASCII)
```

When the stream file appears in the configured source folder, the agent automatically performs the scan.

Benefits of Workflow Integration

Integrating RPG Metrics into the promotion process allows teams to:

- detect complex code earlier in the lifecycle
- identify RPG Free conversion blockers
- highlight high cyclomatic complexity routines
- generate consistent local reports for every promoted program
- maintain dashboard visibility of code quality trends
- keep RPG source code in the environment where scanning takes place

This turns RPG Metrics into a continuous code quality monitoring tool for IBM i development.

Example: MDCMS Integration

When using MDCMS, the `CPYTOSTMF` command can be added to a post-install command when objects are installed in the test environment.

Each promoted object can then automatically trigger a scan without requiring developer interaction.

Distribution

The Agent is distributed as a versioned ZIP package.

Example:

```
rpg-metrics-agent-0.2.4.zip
```

This avoids requiring users to know Git and ensures they receive a clean, tested package.

The ZIP should include:

```
agent.py
agent.conf.example
run-agent.bat
requirements.txt
config/
metrics/
reporting/
README.md
README.pdf
VERSION.txt
```

It should not include:

```
.venv/
__pycache__/
*.pyo
agent.conf
logs/
reports/
sent/
unsent/
```

Ordinary Python cache bytecode must also not be included.

The 0.2.4 package deliberately includes these protected compiled modules:

```
config/features.pyc
metrics/nesting.pyc
```

The corresponding readable source files must not be present in the distributed ZIP:

```
config/features.py
metrics/nesting.py
```

License

RPG Metrics Agent is part of the RPG Metrics platform.

© KDP Software Ltd.